```
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSSSSSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMMMMM   MMMMMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRR         RRR  MMM   MMM   MMM  SSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRRRRRRRRRRR     MMM         MMM     SSSSSSSSS
RRR   RRR        MMM         MMM              SSS
RRR   RRR        MMM         MMM              SSS
RRR   RRR        MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR      RRR     MMM         MMM              SSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
RRR         RRR  MMM         MMM  SSSSSSSSSSSS
```

```
NN      NN  TTTTTTTTTT    000000    EEEEEEEEEE  RRRRRRRR      AAAAAA      SSSSSSSS  EEEEEEEEEE
NN      NN  TTTTTTTTTT    000000    EEEEEEEEEE  RRRRRRRR      AAAAAA      SSSSSSSS  EEEEEEEEEE
NN      NN      TT      00      00  EE          RR      RR  AA      AA  SS          EE
NN      NN      TT      00      00  EE          RR      RR  AA      AA  SS          EE
NNNN    NN      TT      00    0000  EE          RR      RR  AA      AA  SS          EE
NNNN    NN      TT      00    0000  EE          RR      RR  AA      AA  SS          EE
NN  NN  NN      TT      00  00  00  EEEEEEEE    RRRRRRRR    AA      AA    SSSSSS    EEEEEEEE
NN  NN  NN      TT      00  00  00  EEEEEEEE    RRRRRRRR    AA      AA    SSSSSS    EEEEEEEE
NN    NNNN      TT      0000    00  EE          RR  RR      AAAAAAAAAA          SS  EE
NN    NNNN      TT      0000    00  EE          RR  RR      AAAAAAAAAA          SS  EE
NN      NN      TT      00      00  EE          RR    RR    AA      AA          SS  EE              ....
NN      NN      TT      00      00  EE          RR    RR    AA      AA          SS  EE              ....
NN      NN      TT        000000    EEEEEEEEEE  RR      RR  AA      AA  SSSSSSSS    EEEEEEEEEE      ....
NN      NN      TT        000000    EEEEEEEEEE  RR      RR  AA      AA  SSSSSSSS    EEEEEEEEEE      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000            $BEGIN  NTOERASE,000,NF$NETWORK,<NETWORK ERASE FILE>
0000
0000
0000    ;
0000    ;*****************************************************************************
0000    ;*                                                                           *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000    ;*  ALL RIGHTS RESERVED.                                                     *
0000    ;*                                                                           *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER    *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
0000    ;*  TRANSFERRED.                                                             *
0000    ;*                                                                           *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    ;*  CORPORATION.                                                             *
0000    ;*                                                                           *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    ;*                                                                           *
0000    ;*                                                                           *
0000    ;*****************************************************************************
0000    ;
0000
0000    ;++
0000    ; Facility: RMS
0000    ;
0000    ; Abstract:
0000    ;
0000    ;       This module communicates with the file access listener at the remote
0000    ;       node to erase (delete) the specified file.
0000    ;
0000    ; Environment: VAX/VMS, executive mode
0000    ;
0000    ; Author: James A. Krycka,      Creation Date:  09-DEC-1977
0000    ;
0000    ; Modified By:
0000    ;
0000    ;       V03-003 JAK0138         J A Krycka      28-MAR-1984
0000    ;               Call modified NT$EXCH_CNF routine with a parameter.
0000    ;
0000    ;       V03-002 KRM0109         K Malik         12-May-1983
0000    ;               Update to support DAP V7.0 message exchange sequence.
0000    ;
0000    ;--
```

```
0000            .SBTTL  DECLARATIONS
0000
0000      ;
0000      ; Include Files:
0000      ;
0000
0000            $DAPPLGDEF                              ; Define DAP prologue symbols
0000            $DAPHDRDEF                              ; Define DAP message header
0000            $DAPCNFDEF                              ; Define DAP Configuration message
0000            $DAPACCDEF                              ; Define DAP Access message
0000            $DAPNAMDEF                              ; Define DAP Name message
0000            $IFBDEF                                 ; Define IFAB symbols
0000            $NWADEF                                 ; Define Network Work Area symbols
0000
0000      ;
0000      ; Macros:
0000      ;
0000      ;     None
0000      ;
0000      ; Equated Symbols:
0000      ;
0000
0000            ASSUME  DAP$Q_DCODE_FLG EQ 0
0000            ASSUME  NWA$Q_FLG EQ 0
0000
0000      ;
0000      ; Own Storage:
0000      ;
0000      ;     None
0000      ;
```

```
                        0000          .SBTTL  NT$ERASE - PERFORM NETWORK ERASE FUNCTION
                        0000
                        0000    ;++
                        0000    ; NT$ERASE - engages in a DAP dialogue with the remote FAL to erase (delete)
                        0000    ;         the specified file.
                        0000    ;
                        0000    ; Calling Sequence:
                        0000    ;
                        0000    ;         BSBW    NT$ERASE
                        0000    ;
                        0000    ; Input Parameters:
                        0000    ;
                        0000    ;         R8      FAB address
                        0000    ;         R9      IFAB address
                        0000    ;         R10     FWA address
                        0000    ;         R11     Impure Area address
                        0000    ;
                        0000    ; Implicit Inputs:
                        0000    ;
                        0000    ;         FAB$L_NAM
                        0000    ;         IFB$L_NWA_PTR
                        0000    ;
                        0000    ; Output Parameters:
                        0000    ;
                        0000    ;         R0      Status code (RMS)
                        0000    ;         R1-R7   Destroyed
                        0000    ;         AP      Destroyed
                        0000    ;
                        0000    ; Implicit Outputs:
                        0000    ;
                        0000    ;         Resultant name string
                        0000    ;
                        0000    ; Completion Codes:
                        0000    ;
                        0000    ;         Standard RMS completion codes
                        0000    ;
                        0000    ; Side Effects:
                        0000    ;
                        0000    ;         None
                        0000    ;
                        0000    ;--
                        0000
                        0000    NT$ERASE::                              ; Entry point
                        0000          $TSTPT  NTERASE                   ;
     57   3C A9   D0    0006          MOVL    IFB$L_NWA_PTR(R9),R7      ; Get address of NWA (and DAP)
                        000A
                        000A    ;+
                        000A    ; Exchange DAP Configuration messages with FAL and determine DAP buffer size.
                        000A    ;-
                        000A
         50   04  D0    000A          MOVL    #DAP$K_ERASE,R0           ; Denote type of file access
          FFF0'   30    000D          BSBW    NT$EXCH_CNF               ; Exchange Configuration messages
         60 50    E9    0010          BLBC    R0,EXIT                   ; Branch on failure
                        0013
                        0013    ;+
                        0013    ; Next build a request mask (NWA$W_DISPLAY) that will be used in the Access
                        0013    ; message to request that optional DAP messages be returned by FAL. For $ERASE,
                         0013  ; the Name message is the only candidate.
                         0013  ;-
                         0013
                         0013  BUILD_MASK:                                   ; Build NWA$W_DISPLAY
             52    D4    0013          CLRL    R2                            ; Initialize request mask
          FFE8'    30    0015          BSBW    NT$SCAN_NAMBLK                ; Scan user Name Block and check FAL's
                         0018                                                ;  capabilities to update request mask
          58 50    E9    0018          BLBC    R0,EXIT                       ; Branch on failure
 00D0 C7     52    B0    001B          MOVW    R2,NWA$W_DISPLAY(R7)          ; Save request mask
                         0020
                         0020  ;+
                         0020  ; Build and send DAP Access message to partner.
                         0020  ;-
                         0020
                         0020  SEND_ACC:                                     ;
                         0020          $SETBIT #NWA$V_LAST_MSG,(R7)          ; Declare this last message to block
       50    03    D0    0024          MOVL    #DAP$K_ACC_MSG,R0             ; Get message type value
          FFD6'    30    0027          BSBW    NT$BUILD_HEAD                 ; Construct message header
       85    04    90    002A          MOVB    #DAP$K_ERASE,(R5)+            ; Store ACCFUNC field
       85    01    90    002D          MOVB    #DAP$M_NONFATAL,(R5)+         ; Store ACCOPT field
          FFCD'    30    0030          BSBW    NT$GET_FILESPEC               ; Store FILESPEC as a counted
                         0033                                                ;  ASCII string
 51    00D0 C7     3C    0033          MOVZWL  NWA$W_DISPLAY(R7),R1          ; Get request mask
             05    13    0038          BEQL    10$                           ; Branch if nothing set
             85    B4    003A          CLRW    (R5)+                         ; Store dummy FAC and SHR fields
          FFC1'    30    003C          BSBW    NT$CVT_BN4_EXT                ; Store DISPLAY as an extensible field
          FFBE'    30    003F  10$:    BSBW    NT$BUILD_TAIL                 ; Finish building message
          FFBB'    30    0042          BSBW    NT$TRANSMIT                   ; Send Access message to FAL
          2B 50    E9    0045          BLBC    R0,EXIT                       ; Branch on failure
                         0048
                         0048  ;+
                         0048  ; Receive DAP (resultant) Name message from partner.
                         0048  ;-
                         0048
                         0048  RECV_NAM:                                     ;
    00D0 C7        B5    0048          TSTW    NWA$W_DISPLAY(R7)             ; Branch if Name message was not
             1D    13    004C          BEQL    RECV_CMP                      ;  requested
                         004E          $SETBIT #DAP$K_NAM_MSG,DAP$L_MSG_MASK(R7)
                         0053                                                ; Expect response of Name message
          FFAA'    30    0053          BSBW    NT$RECEIVE                    ; Get reply from FAL
          1A 50    E9    0056          BLBC    R0,EXIT                       ; Branch on failure
          FFA4'    30    0059          BSBW    NT$DECODE_NAM                 ; Process resultant name string
                         005C
                         005C  ;+
                         005C  ; Receive DAP Acknowledge message from partner.
                         005C  ;-
                         005C
                         005C  RECV_ACK:                                     ;
       67    26    E1    005C          BBC     #DAP$V_GEQ_V70,(R7),-         ; Acknowledge message required if
             0B          005F                  RECV_CMP                      ;  partner uses DAP since V7.0
                         0060          $SETBIT #DAP$K_ACK_MSG,DAP$L_MSG_MASK(R7)
                         0065                                                ; Expect response of Acknowledge message
          FF98'    30    0065          BSBW    NT$RECEIVE                    ; Get reply from FAL
          08 50    E9    0068          BLBC    R0,EXIT                       ; Branch on failure
                         006B
                         006B  ;+
                         006B  ; Receive DAP Access Complete message from partner to terminate the reply
```

```
                   006B   196 ; sequence.
                   006B   197 ;-
                   006B   198
                   006B   199 RECV_CMP:                                       ;
                   006B   200         $SETBIT #DAP$K_CMP_MSG,DAP$L_MSG_MASK(R7)
                   0070   201                                                 ; Expect response of Access Complete msg
       FF8D'   30  0070   202         BSBW    NT$RECEIVE                      ; Get reply from FAL
               05  0073   203 EXIT:   RSB                                     ; Exit with RMS code in R0
                   0074   204
                   0074   205         .END                                    ; End of module
```

```
NTOI
Syml

$$.I
$$RI
$$RI
$$RI
$$RI
BUII
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
CAP'
CAP'
DAP'
DAP'
DAP!
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP'
DAP
DAP
DAP
DAP
DAP
DAF
DAP
DAP
DAP
DAP
DAP
DAP
DAP
DAP
DAP
DAP
DAP
DAP
```

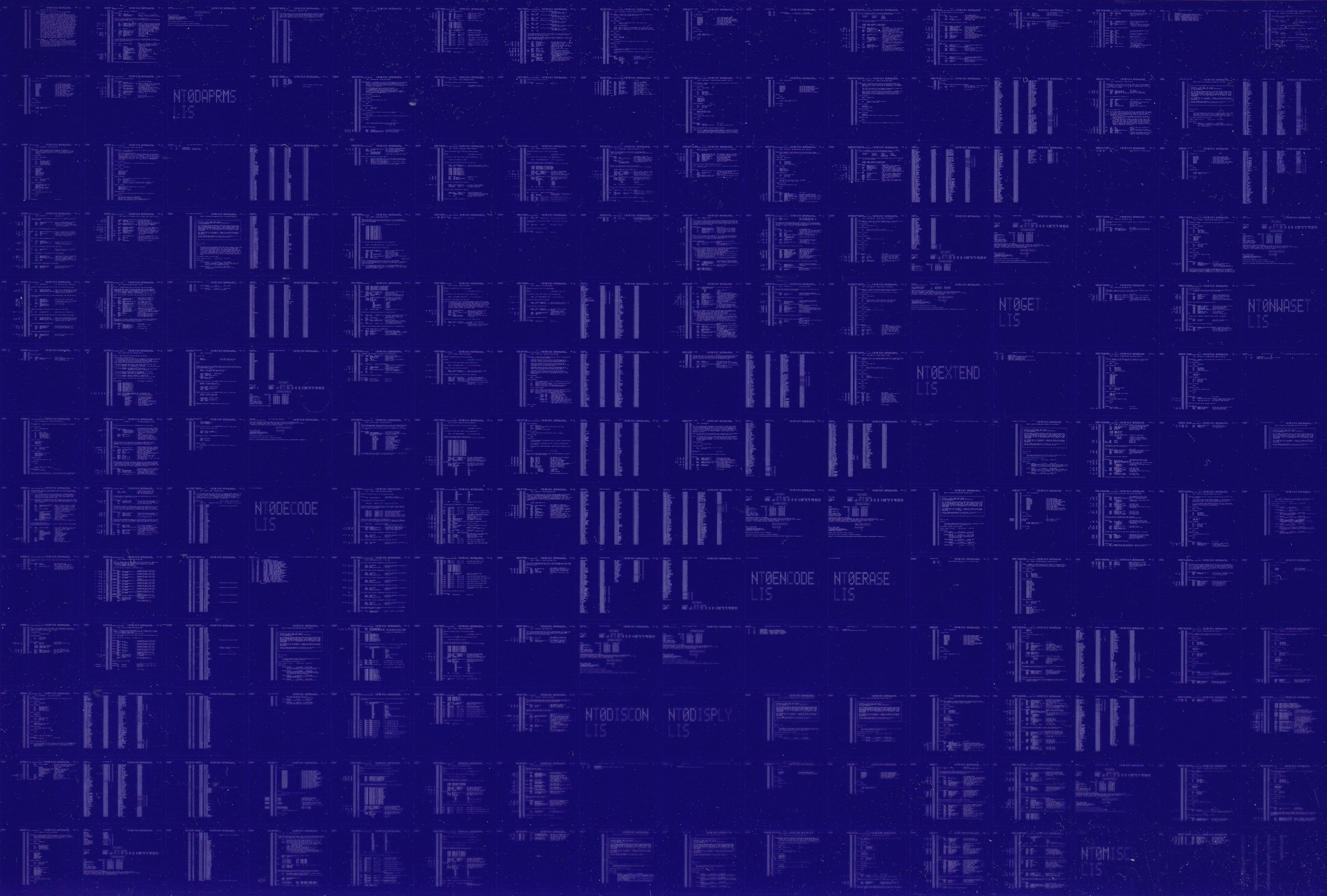
```
$$.PSECT_EP              = 00000000             DAP$Q_SYSCAP         00000028              NWA
$$RMSTEST                = 0000001A             DAP$Q_SYSPEC         00000038              NWA
$$RMS_PBUGCHK            = 00000010             DAP$V_GEQ_V70      = 00000026              NWA
$$RMS_TBUGCHK            = 00000008             DAP$W_BUFSIZ         00000040              NWA
$$RMS_UMODE              = 00000004             DAP$W_DISPLAY1       0000004C              NWA
BUILD_MASK                 00000013 R      01   DAP$W_PARTNER        00000006              NWA
DAP$B_ACCFUNC              00000040             DAP$W_VERSION        00000004              NWA
DAP$B_ACCOPT               00000041             EXIT                 00000073 R      01    NWA
DAP$B_BITCNT               00000035             IFB$L_NWA_PTR      = 0000003C              NWA
DAP$B_DCODE_FID            00000019             NT$BUILD_HEAD        ********    X   01    NWA
DAP$B_DCODE_MAC            0000001B             NT$BUILD_TAIL        ********    X   01    NWA
DAP$B_DCODE_MSG            0000001A             NT$CVT_BN4_EXT       ********    X   01    NWA
DAP$B_DECVER               00000047             NT$DECODE_NAM        ********    X   01    NWA
DAP$B_ECONUM               00000045             NT$ERASE             00000000 RG     01    NWA
DAP$B_FAC                  00000042             NT$EXCH_CNF          ********    X   01    NWA
DAP$B_FILESYS              00000043             NT$GET_FILESPEC      ********    X   01    NWA
DAP$B_FLAGS                00000031             NT$RECEIVE           ********    X   01    NWA
DAP$B_LEN256               00000034             NT$SCAN_NAMBLK       ********    X   01    NWA
DAP$B_LENGTH               00000033             NT$TRANSMIT          ********    X   01    NWA
DAP$B_NAMETYPE             00000040             NWA$B_ALLXABCNT      0000011C              NWA
DAP$B_OSTYPE               00000042             NWA$B_DAP_RAC        000000C9              NWA
DAP$B_SHR                  00000043             NWA$B_FILESYS        000000C5              NWA
DAP$B_STREAMID             00000032             NWA$B_KEYXABCNT      0000011D              NWA
DAP$B_TYPE                 00000030             NWA$B_NETSTRSIZ      0000016F              NWA
DAP$B_USRNUM               00000046             NWA$B_NODBUFSIZ      00000168              NWA
DAP$B_USRVER               00000048             NWA$B_ORG            000000C6              NWA
DAP$B_VERNUM               00000044             NWA$B_OSTYPE         000000C4              NWA
DAP$B_X_FIELD              00000024             NWA$B_RFM            000000C7              NWA
DAP$C_BLN                  000000C0             NWA$B_RMS_RAC        000000C8              NWA
DAP$K_ACC_MSG            = 00000003             NWA$C_BLN            00000800              NWA
DAP$K_ACK_MSG            = 00000006             NWA$K_BLN            00000800              NWA
DAP$K_BLN                  000000C0             NWA$L_ALLXABADR      00000100              NWA
DAP$K_CMP_MSG            = 00000007             NWA$L_DATXABADR      00000104              NWA
DAP$K_ERASE              = 00000004             NWA$L_DEV            000000C0              NWA
DAP$K_NAM_MSG            = 0000000F             NWA$L_FHCXABADR      00000108              NWA
DAP$L_CMWA                 00000030             NWA$L_KEYXABADR      0000010C              NWA
DAP$L_CRC_RSLT             00000020             NWA$L_MSG_MASK       000000D4              NWA
DAP$L_DCODE_STS            00000018             NWA$L_PROXABADR      00000110              NWA
DAP$L_MSG_MASK             0000001C             NWA$L_RDTXABADR      00000114              NWA
DAP$L_SSPQA                00000080             NWA$L_SAVE_FLGS      00000128              NWA
DAP$L_TEMP                 00000090             NWA$L_SUMXABADR      00000118              NWA
DAP$M_BITCNT             = 00000008             NWA$L_THREAD         000000FC              NWA
DAP$M_DFTSPEC            = 00000010             NWA$L_XLTATTR        00000238              NWA
DAP$M_DSP_3NAM           = 00000200             NWA$L_XLTBUFFLG      0000022C              NWA
DAP$M_GET                = 00000002             NWA$L_XLTCNT         00000228              NWA
DAP$M_GO_NOGO            = 00000010             NWA$L_XLTMAXINDX     00000234              NWA
DAP$M_MSE                = 00000010             NWA$L_XLTSIZ         00000230              NWA
DAP$M_NONFATAL           = 00000001             NWA$Q_ACS            00000244              NWA
DAP$M_SEGMENT            = 00000040             NWA$Q_BIGBUF         00000170              NWA
DAP$M_TMP1$              = 00000020             NWA$Q_BLD            000000F0              NWA
DAP$M_TMP2$              = 000000C0             NWA$Q_FLG            00000000              NWA
DAP$Q_DCODE_FLG            00000000             NWA$Q_INODE          0000025C              NWA
DAP$Q_FILESPEC             00000044             NWA$Q_IOSB           000000D8              NWA
DAP$Q_MSG_BUF1             00000008             NWA$Q_LNODE          00000160              NWA
DAP$Q_MSG_BUF2             00000010             NWA$Q_LOGNAME        0000023C              NWA
DAP$Q_NAMESPEC             00000044             NWA$Q_NCB            00000264              NWA
DAP$Q_PASSWORD             00000050             NWA$Q_RCV            000000E0              NWA
```

```
NWA$Q_SAVE_DESC                    00000120
NWA$Q_XLTBUF1                      0000024C
NWA$Q_XLTBUF2                      00000254
NWA$Q_XMT                          000000E8
NWA$T_ACSBUF                       0000026C
NWA$T_AUXBUF                       000005E0
NWA$T_DAP                          00000000
NWA$T_INODEBUF                     000004AC
NWA$T_ITM_ATTR                     00000200
NWA$T_ITM_END                      00000224
NWA$T_ITM_LST                      000002C0
NWA$T_ITM_MAXINDX                  00000218
NWA$T_ITM_STRING                   0000020C
NWA$T_NCBBUF                       0000052C
NWA$T_NODEBUF                      00000169
NWA$T_RCVBUF                       000001A0
NWA$T_SCAN                         00000100
NWA$T_TEMP                         00000120
NWA$T_XLTBUF1                      000002AC
NWA$T_XLTBUF2                      000003AC
NWA$T_XMTBUF                       000003C0
NWA$V_LAST_MSG                   = 00000000
NWA$W_BUILD                        000000D2
NWA$W_DAPBUFSIZ                    000000CA
NWA$W_DIR_OFF                      000000CC
NWA$W_DISPLAY                      000000D0
NWA$W_FIL_OFF                      000000CE
NWA$W_JNLXABJOP                    0000011E
PIO$A_TRACE                        ********    X    01
RECV_ACK                           0000005C R       01
RECV_CMP                           0000006B R       01
RECV_NAM                           00000048 R       01
SEND_ACC                           00000020 R       01
TPT$C_NTERASE                      ********    X    01

                                                 +-----------------+
                                                 ! Psect synopsis !
                                                 +-----------------+

PSECT name                         Allocation          PSECT No.  Attributes
----------                         ----------          ---------  ----------
.  ABS  .                          00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
NF$NETWORK                         00000074  (  116.)  01 (  1.)    PIC   USR   CON   REL   GBL NOSHR   EXE   RD   NOWRT NOVEC BYTE
$ABS$                              00000800  ( 2048.)  02 (  2.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE

                                              +------------------------+
                                              ! Performance indicators !
                                              +------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                       32     00:00:00.11    00:00:01.47
Command processing                  113     00:00:00.57    00:00:05.58
Pass 1                              231     00:00:07.04    00:00:19.43
Symbol table sort                     0     00:00:00.84    00:00:01.04
Pass 2                               56     00:00:01.20    00:00:03.23
Symbol table output                  18     00:00:00.14    00:00:00.37
```

```
Psect synopsis output              1       00:00:00.02     00:00:00.03
Cross-reference output             0       00:00:00.00     00:00:00.00
Assembler run totals             453       00:00:09.93     00:00:31.53

The working set limit was 1200 pages.
33321 bytes (66 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 589 non-local and 5 local symbols.
205 source lines were read in Pass 1, producing 13 object records in Pass 2.
19 pages of virtual memory were used to define 18 macros.

                                              +----------------------------+
                                              ! Macro library statistics !
                                              +----------------------------+

Macro library name                             Macros defined
------------------                             --------------
_$255$DUA28:[RMS.OBJ]RMS.MLB;1                         10
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       4
TOTALS (all libraries)                                 14

794 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:NTOERASE/OBJ=OBJ$:NTOERASE MSRC$:NTOERASE/UPDATE=(ENH$:NTOERASE)+LIB$:RMS/LIB
```

0316 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



NT0DAPRMS
LIS

NT0GET
LIS

NT0NWASET
LIS

NT0EXTEND
LIS

NT0DECODE
LIS

NT0ENCODE
LIS

NT0ERASE
LIS

NT0DISCON
LIS

NT0DISPLY
LIS

NT0MISC
LIS